



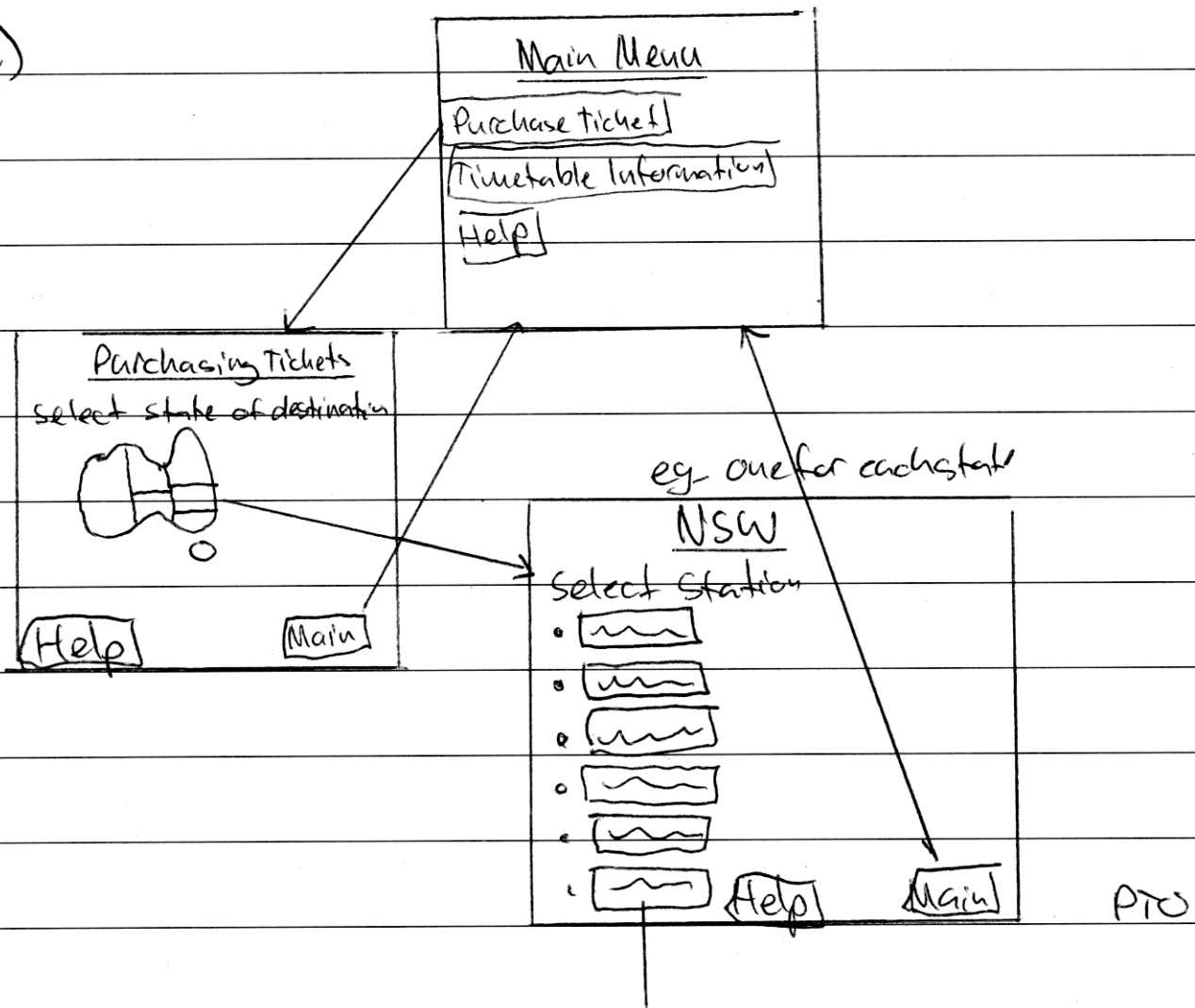
22) a) The most appropriate approach would be the structured approach since it is very large scale project, most likely would have lots of funding and people working on it. It is also fairly crucial that each stage is completed and documentation is thorough as there will be many users, repairmen who will work on it, if it were to malfunction there would be bad publicity which is not desired. Structured approach helps to ensure everything is completed and these problems avoided.

b) Will the touch screens be well built enough to withstand constant and probably damaging use by the general public? This is important as constant repairs will be expensive and not desired, so they must be well built to guard damage.

Is there already hardware available that can

dispense tickets, timetable information etc or will it have to be developed. If it is already developed then it will be far cheaper than attempting to build from scratch.

c)





### Payment

You have chosen to travel to

From

Total cost:

Please select payment option:

Cash

Card

Help

Main

### Cash Payment

Total:

Insert coins into slot

Amount left

Help

Main

### Card Payment

Total

Swipe card

Enter details

Press OK

Help

Main

### Paid

Thank you  
please find your ticket in slot

OK



(i.)

## National Railways

PLEASE SELECT AN OPTION:

Purchase Ticket

Timetable Information

About National Railways

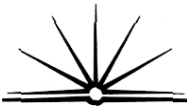
### How to Use This

This is a touch screen system

select an option by lightly

pressing a button with your finger

HELP



d) Blind people would have trouble as it requires observing a computer screen. This could be solved by using sound guides to direct vision impaired people. It would be likely that completely blind people could not fully use this system. The voice prompts should announce screens selected, stations selected, payment details etc.

e) BEGIN MAIN PROGRAM

Print "select destination"

User Dest = User selection

Print "select number single tickets"

Num Single = User selection

Print "select number Return tickets"

Num Return = User selection

indent

⇒ totfare = Calculate Cost (UserDest, NumSingle, NumReturn)



Print "total cost" total fare

END MAIN PROGRAM

BEGIN ~~Sub~~ Calculate Cost (station, Single, Return)

cost = 0

OPEN Station File

index = 0

REPEAT

index = index + 1

UNTIL destination.(index).station = station or index = 100

(if index = 100 ~~then~~ ~~then~~ destination.(index).station  $\neq$  station  
<sup>Print Error no match found</sup> THEN

cost = destination(index).fullsingle \* Single

cost = cost + destination(index).fullreturn \* Return

End ~~Sub~~ Calculate Cost Returns ~~Cost~~ Cost